Understanding visual scenes

Natural Language Engineering **1** (1): 1–22. Printed in the United Kingdom

Understanding Visual Scenes

CARINA SILBERER
DTCL, Universitat Pompeu Fabra,
Roc Boronat 138,
08018 Barcelona, Spain
CarinaSilberer@gmail.com

JASPER UIJLINGS
School of Informatics,
University of Edinburgh,
10 Crichton Street,
Edinburgh EH8 9AB, UK
jrr.uijlings@gmail.com

MIRELLA LAPATA
ILCC, School of Informatics,
University of Edinburgh,
10 Crichton Street
Edinburgh EH8 9AB, UK
mlap@inf.ed.ac.uk

(*Received XX October 2017; revised XX January 2018*)

Abstract

A growing body of recent work focuses on the challenging problem of scene understanding using a variety of cross-modal methods which fuse techniques from image and text processing. In this paper we develop representations for the semantics of scenes by explicitly encoding the objects detected in them and their spatial relations. We represent image content via two well-known types of tree representations, namely constituents and dependencies. Our representations are created deterministically, can be applied to any image dataset irrespective of the task at hand, and are amenable to standard NLP tools developed for tree-based structures. We show that we can apply syntax-based SMT and tree kernel methods in order to build models for image description generation and image-based retrieval. Experimental results on real-world images demonstrate the effectiveness of the framework.

1 Introduction

A growing body of recent work focuses on the challenging problem of scene understanding using a variety of cross-modal methods fusing techniques from image and text processing. Among the tasks dealing with scene understanding, the automatic generation of image descriptions is the most widely studied. Most current approaches draw inspiration from machine translation and use a Recurrent Neural Network (RNN) to learn to "translate" image features into a sentence in English, one word at a time (Vinyals *et al.*, 2015; Xu *et al.*, 2015).

While the results obtained by the RNN-based approaches are often impressive, the learned representations do not explicitly encode the relations between the objects in a scene. They are generally agnostic of *who* is doing *what* to *whom* and as a result are neither interpretable nor particularly suitable for reasoning. Efforts to advance scene understanding have seen the emergence of new tasks such as visual question answering (VQA; Antol et al., 2015), where the aim is to provide an accurate natural language answer given a question about an image, e.g., by predicting the form the answer might take (Kafle & Kanan, 2016). Another strand of research focuses on how to explicitly encode the underlying semantics of images making use of structural representations (Elliott & Keller, 2013; Ortiz *et al.*, 2015; Johnson *et al.*, 2015; Yatskar *et al.*, 2016a). Knowing what entities are depicted in an image and how they relate to each other would allow to correctly infer the actions taking place, the key participants, and their semantic roles.

In this paper, we aim to develop explicit, symbolic representations for the semantics of scenes which we operationalize as visual (constituent or dependency) trees. Our algorithm constructs tree representations deterministically based on the set of objects or entities automatically detected in a scene and their spatial configurations (e.g., `beside, close`). We depart from previous work (Elliott & Keller, 2013; Elliott & de Vries, 2015; Yatskar *et al.*, 2016a) in decoupling the semantic representations from the dataset or application scenario for which they are developed. Our representations are generic, they can be derived for any existing dataset, as long as they contain images, without relying on additional annotator effort or expertise. We also argue that representations based on well-known types of tree structures are expedient from a modeling perspective as they allow us to re-use standard tools and methods from natural language processing. We showcase the utility of our representations in two applications, namely image description generation and image-based retrieval. Our description generation model builds on statistical machine translation (SMT; Koehn et al., 2007), however, unlike related work (Ortiz *et al.*, 2015) which uses a model inspired by phrase-based SMT to produce descriptions for complex abstract scenes (Zitnick *et al.*, 2016), we employ syntax-based techniques and operate on objects automatically detected in real-world images. Moreover, we propose a situation-driven approach that exploits semantic predicate–argument structures to derive training data for components of our generation system. Our image retrieval model uses tree-kernel methods over visual trees to quantify the similarity between images. We experimentally evaluate whether (a) tree-based approaches are superior to models using less or no structure and (b) the type of tree representation (i.e., constituency vs. dependency) has any bearing on model performance. Our results show that both types of representation perform comparably while models making use of bags of objects, relational tuples (Ortiz *et al.*, 2015) or templates (Elliott & de Vries, 2015) lag behind. In comparison with previous work on structural representations of visual scenes (Elliott & Keller, 2013; Johnson *et al.*, 2015), our representation framework is less resource- and labor-intensive. Compared to neural methods, our approach requires more resources but is not task-specific. For example, neural models are typically end-to-end systems developed for a specific task such as image captioning. Our representation framework is more generic—it encodes situations, participants and their spatial relations—and can be easily used for various downstream tasks (not just image captioning) and further reasoning modules.

In the remainder, we first discuss related work and then introduce our visual representa-

tion framework (Section 2). Next, we describe how visual trees lend themselves to the development of image description generation and image retrieval models (Sections 4 and 5). Finally, we present experimental results in Section 6 and conclude the paper in Section 7.

2 Related Work

Previous work has shown that structured representations of images are useful for tasks such as image description generation (Elliott & Keller, 2013; Elliott & de Vries, 2015; Ortiz *et al.*, 2015) and image retrieval (Lan *et al.*, 2012; Elliott & de Vries, 2015). Most related to our research is the work of Elliott & Keller, (2013) who introduce Visual Dependency Representation (VDR) as an intermediate structure that captures the spatial relationships between objects in an image. Follow-up work (Elliott & de Vries, 2015) infers VDRs automatically using an object detector and the description of an image. Descriptions of unseen images are produced by first predicting their VDRs and then generating the text with a template based generation model.

We create visual trees over objects detected in an image deterministically. Our VDR construction procedure is guided by the visual modality alone and is thus applicable to any image dataset with or without descriptions. Aside from dependencies, we also introduce visual representations based on constituency trees and explore differences and commonalities between the two. Our approach is applicable to large-scale datasets exhibiting multiple object classes, actions, and object interactions, while Elliott & de Vries, (2015) focus on 10 actions that are describable by transitive verbs (e.g., *riding a bike*). Other work (Ortiz *et al.*, 2015; Lan *et al.*, 2012) exploits relational tuples of the form object–spatial-relation–object rather than full-blown syntactic representations. Ortiz *et al.*, (2015) use such tuples to represent abstract scenes created from collections of clip art images (Zitnick *et al.*, 2016) and show how these can be used to generate scene descriptions following an SMT-based approach. We experimentally examine whether their model scales beyond abstract scenes and whether it can deal with real-world images and automatically detected objects.

The structured representations we employ capture spatial relationships between objects in an image (e.g., `beside`, `surrounds`). These relationships are typically based on rules which take geometric features into account such as the angle, distance, and pixel overlap of two object regions (Elliott & de Vries, 2015; Ortiz *et al.*, 2015; Yatskar *et al.*, 2016b). Early work on image description generation predicts the prepositional relationships between two objects based on their spatial arrangement. For instance, Kulkarni *et al.*, (2011) use spatial relations to inform the language model of their generation system and for a template-based approach. In a similar vein, Li *et al.*, (2011) create triples encoding a spatial relation between two objects and use the web to find n-grams verbalizing them, while Mitchell *et al.*, (2012) represent spatial relations as prepositions and use them to generate syntactic trees describing an image. Spatial relations have been also used in the context of learning how to manipulate tasks performed by robots such as making a peanut butter and jelly sandwich (Zampogiannis *et al.*, 2015). Finally, in text-to-scene conversion, where 3D scenes are generated from textual descriptions (e.g., *The dinosaur is in front of the horse.*), spatial relations are used to define the basic layout of scenes (Coyne & Sproat, 2001).

Our work joins others in explicitly representing the structure of images in order to enable reasoning about the entities and their relationships. Johnson *et al.*, (2015) introduce scene

graphs which capture the detailed semantics of images by explicitly modeling objects, their attributes, and relationships between objects. They use hand-crafted scene graphs as queries for retrieving semantically similar images (i.e., those images where the query graph can be most likely grounded). Schuster *et al.*, (2015) extend this approach with a parser which maps dependency-parsed natural language descriptions to scene graphs. Other work (Lin *et al.*, 2015) infers 3D scene graphs from images and subsequently converts them to tree representations or employs a knowledge base and reasoning module (Aditya *et al.*, 2016). In comparison, our approach is knowledge-lean, the representations are constructed without recourse to additional annotations or resources such as a database or a parser. Finally, our work relates to recent efforts to define visual semantic role labeling tasks (Yatskar *et al.*, 2016b; Gupta & Malik, 2015) although we do not explicitly identify roles or actions in an image.

3 Structural Visual Representations

In the following we present the details of our visual representation framework which is based on the entities detected in an image and their spatial relations. We first describe the inventory of spatial relations we consider and then discuss our tree construction procedure. Throughout the section we assume we are provided with a set of detected objects; we describe how these are obtained in Section 6.

3.1 Spatial Relations

Table 1 summarizes the definitions we apply for determining a spatial relation between two objects. Our inventory of spatial relations is a refined version of those presented in Elliott & Keller, (2013) and Ortiz *et al.*, (2015) so as to account for the wide variety of object interactions attested in open domain real-world scenes. Elliott & Keller (2013), in conjunction with human annotators, developed the relations with the goal to discriminate between mere object co-occurrence and action-related object interactions. For details, see Elliott (2015, p. 12ff).We introduce relations 3 and 5 and additionally allow combinations of several relations. Parameters in the definitions of relations were based on those deemed optimal in previous work (Ortiz *et al.*, 2015; Elliott & Keller, 2013). Relations 1–6 are mutually exclusive. If one of the conditions for relations 7–8 applies to a pair of objects, we concatenate the relation with the chosen relation from 1–6 to form a finer relation (e.g., `on_below`). We estimate the relative Euclidean distance between the centroids of objects s and t in relation to the length of the image's diagonal (relations 3–6 in Table 1).

We also experimented with learning abstract relations from data through a k-means clustering approach, which used as features three geometric properties, namely pixel overlap of regions, and the angle and the distance between regions (see the definitions in Table 1). Experimental results revealed, however, that tree representations which used these automatically induced relations performed worse experimentally (see the caption generation experiments described in Section 6).

Finally, for an automatic induction of more elaborate and generic relations, which go beyond purely spatial ones, datasets of, e.g., images and their descriptions, would not suffice.

Table 1. *Definitions of spatial relations between objects s and t based on pairwise relationships between bounding boxes B or their centroids C. To compute the angle between centroids, we follow the definition of the unit circle, i.e., $0°$ lies to the east of its center and $90°$ to the north.*

1)	s surrounds t	B_s overlaps at least 90% with B_t.
2)	s on t	B_t overlaps at least 50% with B_s.
3)	s far t	The relative Euclidean distance between C_s and $C_t > 0.72$.
4)	s near t	The relative Euclidean distance between C_s and $C_t > 0.36$.
5)	s veryclose t	The relative Euclidean distance between C_s and $C_t \leq 0.18$.
6)	s close t	The relative Euclidean distance between C_s and $C_t \leq 0.36$.
7)	s below t	The angle between C_s and C_t is between $45°$ and $135°$.
8)	s above t	The angle between C_s and C_t is between $225°$ and $315°$.
9)	s beside t	The angle between C_s and C_t is either between $315°$ and $45°$ or $135°$ and $225°$

This goal would not only require human annotations, but would further lead to arbitrary data (e.g., Johnson *et al.*, (2015))

3.2 Trees

Previous work (Ortiz *et al.*, 2015; Lan *et al.*, 2012) represents an image as a set of binary tuples between objects and their spatial relations. Figure 1 (bottom left) gives an example. Representations based on object pairs cannot express ditransitive verbs. Furthermore tuples with more than two objects are ambiguous with respect to the scope of the spatial relations, unless implicit conventions on the interpretation of their order are applied. For example, in the tuple ⟨PERSON `surrounds` CARROT `on_beside` PERSON `close` GIRAFFE⟩, it is not clear whether the giraffe is close to the person, the carrot, or both.

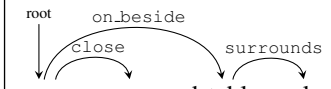
We opt to represent objects and their relations through the use of tree structures (Elliott & Keller, 2013) as they allow us to capture an arbitrary number of scene participants and can express complex relationships involving e.g., transitive and ditransitive verbs. In addition, they allow for the quantification of multiple object class instances (see Figure 1 for an example). In analogy to syntactic representations of sentences, we formulate two different types of visual trees based on dependency and constituency relations. We first formally introduce visual trees and then explain how these are automatically constructed.

Constituency Trees Our visual constituency trees are licensed by a context-free grammar $G_c = \langle R, T, N, S \rangle$. S is the start symbol and T are terminal symbols comprising all spatial relation labels (e.g., `below, surrounds_above`) and object class labels (e.g., UMBRELLA); N denotes non-terminals which in our case are object phrases (NPs; e.g., SKATEBOARD, or SKATEBOARD `on_below` PERSON), spatial relation phrases (SRs;

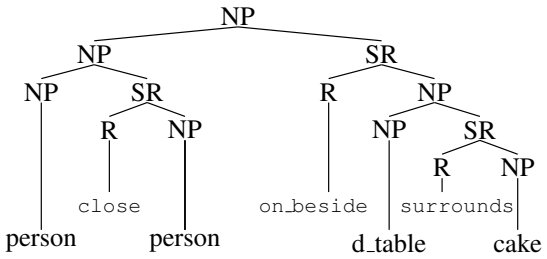
– little kids sitting around a table that has a birthday cake on it
– a group of kids at table around a cake.
– a group of kids around a table with a cake.
– a group of kids watching a woman light a candle.
– a group of young children standing around a cake

person `close` person
person `on_beside` d_table
d_table `surrounds` cake
person `near` cake
(person `close` d_table)
(person `above_close` cake)

TUPLES



DEPENDENCY



CONSTITUENCY

Fig. 1. Top: example of image and its descriptions collected from humans. Tuples in parentheses denote duplicates–both its objects are part of a previous tuple. Bottom: types of structured representations (tuples, dependency, and constituency trees). Spatial relations were determined from image regions (Section 3.1).

e.g., `on_below` PERSON), and spatial relations (Rs; e.g., `on_below`). The grammar rules for G_c are defined as follows:

1. NP $\rightarrow$ NP SR
2. SR $\rightarrow$ R NP
3. R $\rightarrow$ `on` | `below` | `surrounds` | ...
4. NP $\rightarrow$ CAT | UMBRELLA | CUP | ...

Figure 1 (right) shows an example tree.[1] A spatial relation phrase can be interpreted as a stand-in for a verbal or prepositional phrase in a linguistic constituency tree.

Dependency Trees A dependency grammar encodes the structure of sentences by pairwise asymmetric relations between terminal symbols known as dependency (or head-dependent) relations. A dependency structure is a directed acyclic graph which contains the terminal symbols as its nodes, and their dependencies as edges. The (finite) verb is hereby the center of the structure, to which all other nodes are connected through the dependencies. In a visual dependency tree, nodes are object labels and edges correspond to spatial relations. The structural center of the visual dependency grammar is an individual object which is assumed to play a central role in the depicted scene. This definition

[1] dining_table is abbreviated to d_table to save space.

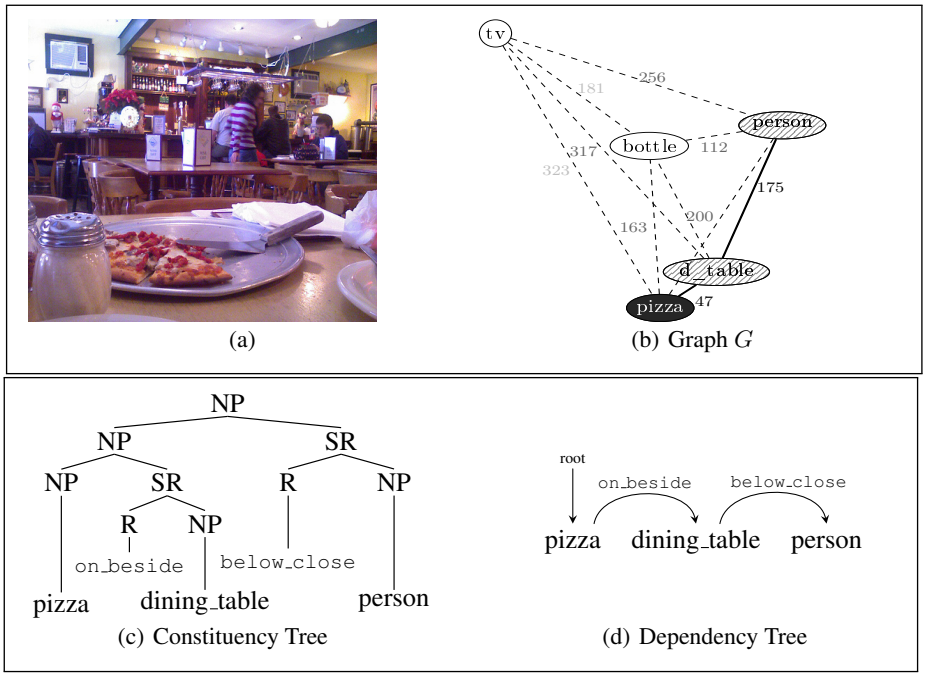


Fig. 2. Illustration of tree creation process. Given a graph G with nodes denoting the objects in an image and edge weights corresponding to their Euclidean distance, we compute a minimum spanning tree for a subgraph $G' \subseteq G$ (filled nodes and plain edges in (b)) and use it to build a constituency (c) and dependency (d) tree, starting from an individual node in the spanning tree (black node in (b)).

is similar to Elliott and Keller's (2013) VDR, however our trees are ultimately different due to another construction procedure which we outline below. The dependency grammar G_D in Table 1 defines the set of spatial relations we employ and Figure 1 (bottom left) provides an example of a visual dependency tree, with the center (PERSON) being the dependent of the root label. Dependencies can capture nuanced semantics of seemingly related images. For example, an image showing *a man standing on a flying airplane* is representated by `below_veryclose`(AIRPLANE, PERSON), whereas the tree for an image depicting *people boarding a plane* would be `surrounds`(AIRPLANE, `above_veryclose`(PERSON, PERSON)). Alternatively, if the image showed *people standing by an airplane* the appropriate representation would be `veryclose`(AIRPLANE, `above_veryclose`(PERSON, PERSON)).

Visual dependency trees capture object interactions via pairwise spatial relationships. The latter are part of the grammar which describes the language. Constituency trees explicitly group objects in spatial proximity into phrases. Spatial relations and object labels are both terminal symbols, i.e., words of the language, and differ in terms of their grammatical categories (formally, NP and R). Intuitively, constituency trees may be better suited at modeling phrases (i.e., groups of objects) and their compositionality.

3.3 Tree Creation

For each image, we construct a tree-based representation automatically in a deterministic manner. Assuming objects have been identified with an object detector, we rank all possible object pairings in ascending order of the Euclidean distance between their centers. Starting with a forest where each node (i.e., object) is a separate tree, we create a minimum spanning
NEW tree (MST). **A MST is a subset of the edges of a connected, edge-weighted graph which connects all the nodes of the graph together and whose sum of edge weights is as small as possible.** We process the list of object pairs top-down, connecting the corresponding nodes if they are not yet in the same tree, until all objects have been linked. We then transfer
NEW the MST to a constituency or dependency tree. **In our MST construction procedure we follow Kruskal's (1956) MST algorithm.**[2] We use this algorithm due to its simplicity and general applicability to a variety of graph structured problems, however other well-known algorithms (Prim, 1957; Karger *et al.*, 1995) could have been employed instead.

Figure 2 illustrates the construction process. The five objects identified in image (a) constitute nodes in graph (b); graph edges denote spatial proximity; as mentioned earlier edge weights correspond to the Euclidean distance between their objects' centers. Filled nodes in the graph represent the MST which can be straightforwardly converted to a dependency tree as follows: we first select the root node according to a criterion such as the prediction score of the object detector, or the top-ranked object as determined by a content selection model (see Section 4.2). This node then determines the direction of all edges which we label with their respective visual relations. For example, to construct the dependency tree shown in Figure 2 (d), we first link the object PIZZA which has been deemed the central object (see Subfigure (b)) to the root node. We next insert the node DINING_TABLE as a dependent of PIZZA, a nd use the definitions in Table 1 to compute their spatial relation and link the nodes with a corresponding edge (`on_beside`). DINING_TABLE, is in turn linked to PERSON in the MST, and we insert the node and the corresponding edge applying the same procedure as before.

We also construct constituency trees based on the MST. We use grammar G_c (defined in Section 3.2) and build a tree with a deterministic bottom-up parse. As the leftmost node of the tree we select n_o from the edge $\langle n_o, n_i \rangle$ with the least weight in the MST (e.g., NP(pizza) in $\langle$NP(pizza), NP(dining_table)$\rangle$ in Figure 2). This node is again selected according to some criterion, as explained in the dependency tree creation. We use the definitions in Table 1 to compute the spatial relation between the nodes n_o and n_i, and call rules 3 and 4 of grammar G_c (in the example in Figure 2, this results in NP(pizza), R(`on_beside`), NP(dining_table)). Rule 2 of G_c can then be applied, which reduces the corresponding spatial relation and n_i to a spatial relation phrase (e.g., SR(R(`on_beside`), NP(dining_table))). Next, rule 1 from G_c reduces SR and the NP which dominates node n_o to an NP (e.g., NP(NP(pizza), SR(R(`on_beside`), NP(dining_table)))). Since no other rules apply, the tree creation proceeds with the next edge $\langle n_j, n_k \rangle$ of the MST which has the least weight and is connected to the already processed nodes (e.g., $\langle$dining_table, person$\rangle$,

[2] More formally, Kruskal's algorithm takes the following steps: 1. Create a set $\mathcal{F}$ of N trees (forest), where each tree is a single node. 2. Sort all the edges in non-decreasing order of their weight. While the list of edges is not empty and $\mathcal{F}$ has less than $N - 1$ edges, repeat step 3: Pick the edge with minimum weight. If it connects two different trees, add it to $\mathcal{F}$. Discard it otherwise.

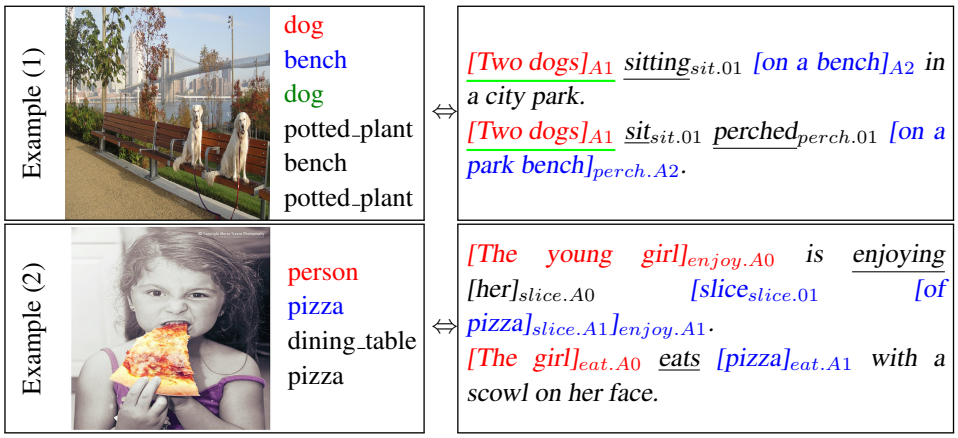


Fig. 3. Example of predicate–argument structures grounded in the images by finding alignments (colored) between the detected objects (left) and the arguments (right).

with $n_j == n_i$). It calls rules 3 and 4 (e.g., R(`below_close`), NP(person)), and rule 2 (e.g., SR(R(`below_close`), NP(person))). Again, rule 1 is called to reduce the NP, which dominates node n_j, and the SR to an NP (e.g., the tree in Figure 2). Tree creation continues until all edges have been processed.

In sum, we use the MST to guide our tree creation procedures. The MST is derived on the basis of the Euclidean distance between the objects' centers, and as a result a tree's adjacency relations between objects reflect closeness in the visual modality—objects that are in a direct head-dependent relationship in a tree appear relatively close to each other in the image.

4 Image Description Generation

We will now illustrate how the proposed tree-based representations can be used to generate image descriptions. The task has recently received significant attention due to the creation of large datasets containing images and associated descriptions (Chen *et al.*, 2015) and new learning techniques based on multimodal recurrent neural networks (Vinyals *et al.*, 2015; Xu *et al.*, 2015). A typical generation model is trained on a parallel corpus of image-description pairs. Since in our case, images are represented by visual trees, we can re-purpose statistical machine translation (SMT) machinery for this task. Specifically, we train a syntax-based tree-to-text SMT system where the source language is rendered as a tree (see Section 3) and corresponds to the visual modality. The target language is the textual modality, i.e., natural language verbalizations of the objects or entities depicted in the image and their interactions.

To describe a new image i, we first perform content selection, i.e., we determine the relevant objects $\mathcal{O}(i)$ from the overall set of detected objects. We then convert $\mathcal{O}(i)$ to visual representations, as explained in the previous section, and use the SMT model to decode the visual content of i to natural language. In the following we describe our image description generation model in more detail.

4.1 Parallel Corpus Creation

The training of an SMT model requires a parallel corpus which consists of pairs of descriptions and visual trees, where each object or entity in the tree is mentioned in the description, and each situation or event can be grounded in the image. Such grounding is absent from most existing datasets (but see Krishna et al., 2016; Plummer et al., 2015, for exceptions) and must be somehow inferred. We detail below how we obtain this information on the Microsoft COCO Captions dataset (Chen *et al.*, 2015) which we use in our experiments (see Section 6 for details), but our approach is general and can be applied to any corpus of images and descriptions.

We preprocess the descriptions with a semantic role labeler and align the identified semantic arguments with the objects detected in the image. Specifically, we parse the descriptions with PathLSTM, a state-of-the-art SRL system which is based on dependency path embeddings (Roth & Lapata, 2016). We extract all predicates (e.g., *eat*) along with their arguments (e.g., constituents *the girl* and *pizza* bearing labels A0 and A1[3] in Figure 3, Example (2)) and modifiers.

Let A denote a predicate–argument structure extracted from the description of image i containing objects $\mathcal{O}(i)$ (e.g., person in Figure 3, Example (2)). To ground A in image i, we find an alignment α which links each argument $a \in A$ (with $n = |A|$) to its referring object(s), $s \in \mathcal{O}(i)$. Function (1) below scores the alignment between object s of class $l(s)$ and argument a:

$$\mathrm{score}_{\mathrm{al}}(s, a) = \mathrm{e}(l(s)|a)\,\mathrm{d}(s), \tag{1}$$

where $\mathrm{d}(s)$ is the class detection score of s, and $\mathrm{e}(l(s)|a)$ the probability of class $l(s)$ being aligned to argument a. If there is a direct string match between the object class (or its category, aka hypernym) and the argument, we set the probability e to 1. Otherwise, we estimate e from object-noun alignments which we obtain by running giza++ (Och & Ney, 2003) on a corpus containing pairs of object class labels detected in an image and all nouns found in its corresponding description.

Let σ denote a permutation on a subset of $\mathcal{O}(i)$, and $|a|$ the minimum number of a's referents (e.g., $|a| = 2$ for *two dogs*, *men*). An alignment α is obtained by the permutation on the object-argument pairings $(\sigma(s_1), a_1), \ldots, (\sigma(s_{m+n}), a_n)$, $m = \sum_{j=1}^{n}(|a_j| - 1)$, which maximizes the sum of the individual alignment scores:

$$\alpha^* = \max_{\sigma} \frac{1}{n} \sum_{j=1}^{n} \sum_{k=0}^{|a_j|-1} \frac{1}{|a_j|} \mathrm{score}_{\mathrm{al}}(\sigma(s_{j+k}), a_j) \tag{2}$$

Overall, we keep those alignments which link each argument of a predicate–argument structure and have a score greater than a threshold β (tuned experimentally on validation data). We leave the linking of modifiers optional. Figure 3 shows two examples of images, their descriptions annotated with predicate–argument structures, and the established alignments.

Grounded predicate–argument structures are paired with simplified image descriptions. We combine the surface strings of the predicate and the aligned arguments and remove

[3] A0 labels denote agents, while A1 labels denote patients or themes (Palmer *et al.*, 2005).

any ungrounded components of the arguments, including adjectives or noun adjuncts. For instance, in Figure 3, for Example 2, we create the sentence *the girl is enjoying her pizza* for the predicate *enjoying* and its arguments *A0* and *A1*, and in Example (1) we build the descriptions *two dogs sit* and *two dogs perched on a bench*. We furthermore create descriptions for the content captured by multiple grounded predicate–argument structures, such as in Figure 3 (Example 1) for which we obtain the description *two dogs sit perched on a bench*. Finally, by applying simple rules based on the dependency parses and the PoS-tags of the captions, we create additional descriptions with alternative referring expressions for quantity groups and for noun phrases with coordinating conjunctions. For example, in the description *a group of people are gathered around a table* we replace the subject *a group of people* by its semantic head, i.e., *people*, which results in the alternative *people are gathered around a table*. And we split the caption *a table that has some food and a drink* into two alternatives: *a table that has some food* and *a table that has a drink*. As a result, it is possible to obtain multiple descriptions for every original caption of an image. Note that in the parallel corpus each visual tree which corresponds to an individual caption is built such that it only contains the objects mentioned in the caption. In other words, the semantic content of each caption is licensed by the tree and vice versa. And since we construct the corpus on the basis of predicate-argument structures, this information corresponds to the verbalization of the objects' interactions or relationships.

4.2 SMT Model

Surface Realization SMT formulates the problem of translating from a source to a target language as finding the translation t that maximizes the conditional probability:

$$t^* = \arg\max_t p(t|s) \tag{3}$$

where $p(t|s)$ is approximated by a conditional log-linear model. Syntax-based SMT (Huang, 2006) defines $p(t|s)$ in terms of the probabilities of individual derivations $d \in D(s,t)$ in a synchronous grammar. The objective function is typically:

$$d^* = \arg\max_d \left(\sum_{k=1}^{K} \lambda_k h_k(d) \right) \tag{4}$$

where h_k denote feature functions which define different models, such as a language model, a translation table and a word penalty model. The constants λ_k are tuned during training and used to scale the different models. In our experiments, we used the tree-to-string SMT framework implemented in Moses (Koehn *et al.*, 2007) to train our models.

Content Selection At test time, we must decide what to talk about, i.e., which objects to focus on. We use logistic regression with l_2-regularization and one hidden layer to predict whether a detected object s is likely to be relevant for the scene, while taking into account other detected objects. This way we do not impose any restrictions on the number of relevant objects, and allow tree-based representations of arbitrary size. The regression model is trained on positive and negative instances obtained from the parallel corpus described in the previous section.

We create a positive training example for each object which was aligned to an argument

in the SRL-parsed image descriptions. As negative examples we select objects with classes different from those of the aligned objects. We represent objects as vectors using a mixture of unary and binary features. The former include the detection score D_t of any object t, and the relative size S_s of target object s in relation to the average relative object size of its class:

$$S_{s_i} = \frac{area(s_i)}{area(i)} / \left[\frac{1}{|O(l_{s_i})|} \sum_{t_j \in O(l_{s_i})} \frac{area(t_j)}{area(j)} \right] \quad (5)$$

where $area(s_i)$ is the area of object s_i in image i. S_s is higher if object s (rendered by its bounding box) is larger than we would expect for its object class l_s, which we estimate from all instances $O(l_s)$ of class l_s in our training data (see Section 6.1 for details). Intuitively, S_s is a means to express the salience of an object in a specific scene. Binary features include the relative distance $D_{s,t}$ between objects s and t, and their normalized co-occurrence frequency $F_{s,t}$. We estimate $D_{s,t}$ as the relative Euclidean distance between the centers of s and t (in relation to the length of image i's diagonal). $F_{s,t}$ is estimated from the object occurrences in a parallel visual-linguistic corpus. We furthermore include the spatial features used for the determination of the spatial relation between two objects, i.e., the relative area of s's bounding box (unary), the intersection-over-union (IoU) of s and t, and the normalized angle between s and t.

5 Query-by-Example Image Retrieval

In query-by-example image retrieval (also called content-based image retrieval), images are retrieved from a collection so that they are maximally similar to some query image. NEW **The similarity of images, however, is ambiguous and depends on the user intent and the purpose of the query. A large body of work has focused on images of city landmarks (e.g., Oxford or Paris (Philbin *et al.*, 2007; Philbin *et al.*, 2008)) or of holiday scenes, such as water effects (Jégou *et al.*, 2008). Since our goal, in contrast, is to capture the semantic content of visual scenes with respect to object interactions, we follow previous work along this line (Elliott *et al.*, 2014) and define image similarity in terms of the actions (e.g., *hit*) or states (e.g., *sit*) they depict. Specifically, the task is to, given a query image showing an action or state, find all images which depict the corresponding predicate (Elliott *et al.*, 2014).**

More formally, let $\mathcal{I}$ denote an image collection. For every query image q_i, we produce a ranking of all images in $\mathcal{I}$ in the order of their similarity to q_i. We estimate the similarity of structured visual representations using tree kernels. The latter have been applied to several NLP tasks ranging from syntactic parsing (Collins & Duffy, 2001), to predicate argument classification (Moschitti, 2006b), and relation extraction (Culotta & Sorensen, 2004). Tree kernels have proven a practical alternative to feature-based methods since they do not require the conversion of trees and their substructures into feature vectors. Instead, they allow implicit exploration of the entire feature space of substructures and compute the similarity between two trees in terms of the number of common tree fragments (Collins & Duffy, 2001).

We compute the similarity of two trees T_1 and T_2 via K, a tree kernel which measures the number of common fragments between T_1 and T_2, with a decay factor of λ (see Mos-

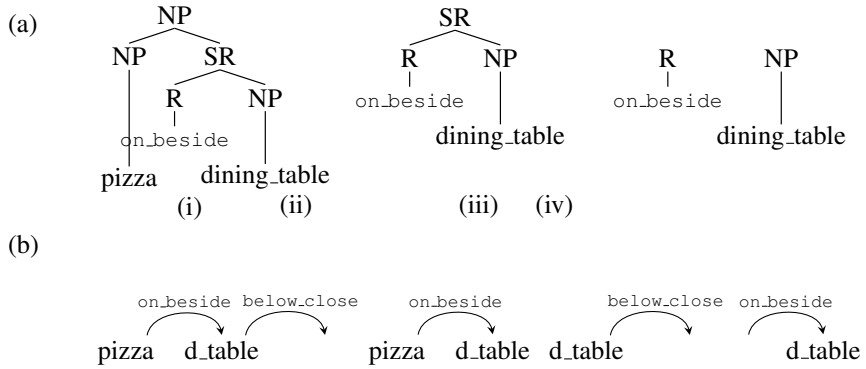


Fig. 4. Subtrees (a) and partial trees (b) of the visual constituency and dependency trees, respectively, shown in Figure 1.

chitti 2006a for details). We additionally apply normalization to obtain a score between 0 and 1:

$$K'(T_1, T_2) = \frac{K(T_1, T_2)}{\sqrt{K(T_1, T_1) \times K(T_2, T_2)}} \tag{6}$$

We use a subtree (ST) kernel to measure the similarity of two images represented by visual constituency trees. A ST is a tree fragment in the constituency tree which may consist of any tree node and its descendants including the leaves (Vishwanathan & Smola, 2003).[4] Figure 4 (a) shows four (out of eight) subtrees of the constituency tree in Figure 2. For the sake of simplicity, we exemplify the similarity estimation with ST kernels by means of the two first subtrees $T_{(i)}$ and $T_{(ii)}$ in Figure 4(a). The common subtrees are $T_{(ii)}$ to $T_{(iv)}$, hence $K(T_{(i)}, T_{(ii)}) = 3$ (with $\lambda = 1$). Applying Equation (6) gives $K'(T_{(i)}, T_{(ii)}) = 0.77$, since $K(T_{(ii)}, T_{(ii)}) = 5$ and $K(T_{(i)}, T_{(i)}) = 3$.

For visual dependency trees we follow Moschitti, (2006a) and use partial tree (PT) kernels. A PT is any substructure of a tree — in our case PTs also contain nodes referring to spatial relations. Examples of PTs are shown in Figure 4(b)[1]. The similarity of trees with PT kernels is estimated in the same fashion as illustrated for ST kernels.

6 Experiments

We present experiments on image description generation and image retrieval. In both cases we address the question of whether the tree structured representations introduced in this paper are beneficial for capturing the semantic content of images. We first describe the datasets used for our experiments, preprocessing tools, and comparison systems. We then present our results on both tasks, and conclude this section with an error analysis.

[4] More powerful kernels which consider all subset trees resulting from a grammar rule (e.g., `SR(R NP)`) are less appropriate in our case due to the small size of our grammar.

6.1 Experimental Setup

Datasets For both tasks we used the publicly available Microsoft COCO Captions dataset[5] (Chen *et al.*, 2015) which contains more than 200,000 images (80K training, 40K validation, and 80K testing images). All training and validation images are annotated with multiple object instances from 80 object classes (e.g., HORSE, AIRPLANE, BANANA), their categories (e.g., ANIMAL, VEHICLE, FOOD), and five captions, inter alia. The captions were collected via Amazon's Mechanical Turk (AMT) by asking annotators to describe the important parts of an image. We used the COCO training set in order to create a parallel corpus as described in Section 4.1 for training the SMT-based description generation models and the content selection model. In total, our training data contains approximately 50K instances. The mean depth of the trees is 3.5 (std=0.99; constituency) and 3.3 (std=0.85; dependency).[6] On average, the training descriptions verbalize the interactions between two and three objects.

The reference descriptions for the COCO test data are not publicly available, and the performance of a model on the test data can be benchmarked by uploading its generated captions to the COCO test server. Note, however, that only automatic evaluation metrics, such as BLEU (Papineni *et al.*, 2002), are used, which compare reference and system captions automatically on the basis of, e.g., n-gram agreement. Since we want to evaluate the models against human judgments in addition to reporting results using automatic measures, we created a test set of reference captions for N = 208 images from the COCO 2015 test set. In view of our interest in the verbalization of the relation between objects, images were sampled so that they contained highly frequent (70% of all test instances), moderately frequent (15%), and infrequent object pairs (15%; see Figure 5 for examples). Following the AMT experimental design described in Chen *et al.*, (2015), we elicited five reference descriptions for each of the sampled images.

NEW **There is a long list of datasets created on top of the COCO images for tasks such as visual question answering (VQA, Antol *et al.*, (2015)) or referring expressions interpretation (RefCOCO, Yu *et al.*, (2016)). As far as content-based image retrieval (CBIR) is concerned, however, to the best of our knowledge, a corresponding extension of COCO does not exist. Elliott et al.'s (2014) work on predicate-centered CBIR, in turn, was limited to only ten actions. As we will explain along the discussion of our experimental results, we therefore created a dataset from COCO by extracting the predicates from the image descriptions.**

Preprocessing We used Girshick's (2015) fast region-based convolutional neural network (Fast R-CNN) in order to detect objects in images. The software takes as input the whole image and region proposals and classifies and refines the proposals. Specifically, for each proposal it outputs a softmax probability distribution over $K + 1$ object classes as well as refined bounding box positions for every class. We used the publicly available pre-trained ImageNet (Russakovsky *et al.*, 2015) model CaffeNet[7] and fine-tuned it on the 80 object

[5] `http://mscoco.org/`
[6] For technical reasons, we represent edge labels by separate nodes, i.e., a dependency tree with two objects has depth 3.
[7] Available at `https://github.com/BVLC/caffe/wiki/Model-Zoo`

Fig. 5. Examples of images with frequent (left), moderately frequent (middle), and infrequent (right) object pairs from the COCO 2015 test set.

classes of the COCO object detection challenge using Fast R-CNN. Object region proposals were extracted with selective search (Uijlings *et al.*, 2013). We filtered overlapping bounding boxes with a symmetric IoU of more than 50%. Finally, we extracted the top 15 detected locations from the images and filtered overlapping boxes rendering the same object class with a 0.3 IoU threshold.

6.2 Comparison Models

Bag-of-Objects This baseline makes use of detected objects only, and does not exploit any structural information. The visual representation of an image is simply a bag of the top-N predicted objects. To evaluate this model on the image retrieval task, we create a flat tree by linking each object to a root node. For the image description task, we train an SMT model on a parallel corpus of bag-of-objects[8] and COCO descriptions.

Tuples Our second comparison model is the approach put forward in Ortiz *et al.*, (2015) who represent visual scenes via tuples (see the example in Figure 1). Content selection in their model is implemented as an integer linear program (ILP) which selects description-worthy pairs of objects using features such as their relative distance $D_{s,t}$ and their normalized co-occurrence frequency $F_{s,t}$ (see Section 4). To generate descriptions with this model, we adapt the pipeline described in Section 4.2, to the tuple-based framework. We create parallel data of visual tuples and descriptions from our cross-modal corpus of object-argument alignments and use it to train a phrase-based SMT model. We then run the ILP on each test image to obtain relevant object pairs whose interactions we wish to describe, convert them to visual tuples, and use the SMT model to transfer them to their verbalizations. For the image retrieval task, we transfer the tuples to trees in which two objects are direct children of the spatial relation that holds between them. Tree kernel methods are then used to measure image similarity.

Templates In our description generation experiments, we further compared against the template-based model of Elliott & de Vries, (2015) which generates a description for an image by filling in the template "DT o_s *is* V DT o_t". The slot DT takes a determiner (*a* or *an*), o_s and o_t are filled with the class labels of the parent and child object nodes in the corresponding visual tree representation (e.g., o_s = DOG and o_t = FRISBEE). Slot V is

[8] We set $N = 2$, as determined on the COCO validation set.

filled with the verb v that maximizes the conditional probability:

$$v^* = \arg\max_v p(v|head, child, sr),$$

where sr is the spatial relation between the objects o_s and o_t (e.g., $p(\text{catch}|\text{DOG}, \text{FRISBEE}, \texttt{very\_close})$, $p(\text{eat}|\text{PERSON}, \text{PIZZA}, \texttt{surrounds\_above})$). We used the parallel corpus described in Section 4.1 to derive these probability estimates. In cases where we cannot find a verb to fill slot V, we default to the spatial relation sr (e.g., surrounds or is near).

NIC Finally, we compare our approach to description generation against NIC, the Neural Image Caption model of Vinyals *et al.*, (2015).[9] We selected NIC as an example of RNN-based approaches. It was also trained on COCO and performs reasonbaly well even against a nearest neighbor baseline (Vinyals *et al.*, 2015). The latter baseline has proven very competitive on COCO due to the images being similar and as result giving rise to similar linguistic descriptions (Devlin *et al.*, 2015). In Vinyal et al.'s (2015) approach, a convolution neural network (CNN) encodes an image into a compact representation, which is then fed to a long short-term memory (LSTM) decoder to generate a sentence. The LSTM is trained to maximize the likelihood of the correct description given the image.

VGG For our image retrieval experiments, we employ a nearest-neighbor (NN) approach (Devlin *et al.*, 2015) which finds the k NNs using cosine similarity and represents images with a 4096-dimensional vector. The latter is extracted from the image activations from the penultimate layer (fc7) of a pre-trained CNN model VGG-16 (Simonyan & Zisserman, 2014).

6.3 Parameter Settings

All parameters, including the choice of spatial relations, the features for the content selection models, and the parameters for grounding predicate–argument structures, were tuned on the COCO validation data. All SMT models were trained on the corresponding parallel corpus obtained from COCO. 6-gram SMT language models were trained on both Flickr30K descriptions (Young *et al.*, 2014) and COCO descriptions with modified Kneser-Ney smoothing using KenLM (Heafield *et al.*, 2013).

6.4 Results

6.4.1 Image Description Generation

We measured the quality of the generated descriptions against reference captions using BLEU (Papineni *et al.*, 2002) and CIDEr, a consensus-based metric developed specifically for image description evaluation (Vedantam *et al.*, 2014).[10] We further evaluated system

[9] We used Karpathy's publicly available implementation at https://github.com/karpathy/neuraltalk.
[10] Using the tools provided at http://mscoco.org/.

Table 2. *Model comparison on 208 COCO 2015 test images using automatic measures (B is a shorthand for BLEU). All models except Templates and NIC are SMT-based generation models, Bag-of-Objects and NIC do not use visual relations.*

Model	CIDEr	B1	B2	B3	B4
Bag-of-Objects	44.1	47.9	28.0	15.4	8.3
Template	43.8	55.5	30.7	14.9	6.3
Tuples	47.9	53.2	34.5	17.8	9.9
Dependency	54.3	**59.1**	**40.1**	**23.2**	13.0
Constituency	52.0	55.1	36.9	21.5	**14.0**
NIC	**58.8**	54.0	34.4	21.3	13.3

output against human judgments which we elicited for our COCO test fraction using AMT. Participants were presented with an image (9 per task) and seven descriptions (two by our models, four by the comparison models, and one randomly selected reference description). They were asked to rank the descriptions from best to worst (1–5, ties were allowed) in order of informativeness and grammaticality. Participants were instructed to give low ranks (ranks 1 or 2) to descriptions which were grammatical and faithfully described the content of an image, and to penalize descriptions which focused on un-important aspects. For every image we collected 10 ratings.

Table 2 summarizes the results of the automatic evaluation. Overall, we observe that Template, the only system that does not apply SMT methods, performs worst. Tuples, which exploits visual relationships, is more effective than Bag-of-Objects, which is based on object labels only. And the tree-based models, Dependency and Constituency, outperform all other symbol-based models. This indicates that structured representations exploiting visual relationships between objects are beneficial for verbalizing their interactions. NIC performs best according to the CIDEr metric. Although it does not explicitly encode spatial relationships, it benefits from its high-dimensional feature representations for visual and language content (i.e., CNN-based feature vectors for images and objects occurring in them, and text embeddings).

The results of the human evaluation study follow a similar pattern in Table 3. Dependency and Constituency are more often judged best or second-best (i.e., ranks 1 or 2) than the other symbol-based models (27% and 29% of all generated descriptions, respectively), while NIC seems to have a lead. In most cases (72% of time) the output of Bag-of-Objects is judged second-worst or worst (i.e., ranks 4 or 5). None of the models, however, comes close to the human upper bound, where 77% of the sentences are judged best or second best (last row in Table 3). Pairwise differences between all systems are statistically significant (using post-hoc Tukey tests; $p < 0.01$), except for Constituency and Dependency.

Table 3. *Rankings given to systems by human subjects on 208 COCO 2015 test images. Rankings are shown as proportions of the total number of ratings per model (columns 2–4) and grouped into top (ranks 1–2), middle (rank 3), and bottom (ranks 4–5). The numbers do not sum to one as ties are allowed. The last column gives the mean rank computed across all annotators and images (lower is better).*

Model	1–2	3	4–5	∅ rank
Bag-of-Objects	0.14	0.13	0.72	4.1
Template	0.20	0.14	0.65	3.8
Tuples	0.23	0.15	0.62	3.7
Dependency	0.27	0.16	0.57	3.6
Constituency	0.29	0.15	0.56	3.6
NIC	0.39	0.14	0.46	3.2
Humans	0.77	0.11	0.12	1.8

Table 4. *Macro-averaged precision on 868 COCO validation images and 61 verb types. Numbers in parentheses are micro-average.*

Model	P@5	P@10
Bag-of-Objects	13.7 (22.9)	12.3 (21.1)
Tuples	12.6 (20.2)	11.9 (19.4)
Constituency	14.5 (22.4)	13.1 (21.6)
Dependency	16.2 (24.7)	15.6 (23.6)
VGG	19.1 (32.3)	16.6 (30.8)

6.4.2 Image Retrieval

We evaluated model performance using precision at rank k (P@k), the proportion of images among the top k which are annotated with the same predicate as the query image q_i. We used the MSCOCO 2014 validation dataset as our image collection $\mathcal{I}$, and extracted the verbal predicates which occurred at least in two descriptions per image, and were mentioned in at least five images. This resulted in 61 verb types associated with 868 images in total.

Table 4 gives results on the COCO validation set $\mathcal{I}$ with each image as query. Table 5 shows the results on the test set (208 images); we report P@1 in addition to P@5 and P@10. The two tree-based representations, Dependency and Constituency, are the overall

NIC	4 a person on a surfboard in the water	3 a couple of giraffe standing next to each other	6 a man riding a skateboard down the street
Temp	4 a person is holding a person	2 a person is feeding a giraffe	3 a person is above a suitcase
Tuples	5 a man holding a man	2 a person feeding a giraffe	4 a man holding a suitcase
Dep	3 a bear	1 a person feeding a giraffe	2 a suitcase on the street
Const	2 a person is watching a bear	1 a man feeding a giraffe	5 a man with luggage
Human	1 a large bear swimming in a pool of blue water	1 a giraffe is eating out of a person's hand	1 a woman drags a suitcase down a street

NIC	4 a close up of a plate of food on a table	6 a man is holding a bunch of bananas	4 a young boy is eating a piece of cake
Temp	6 a cup is below a cup	3 an apple is near an apple	2 a person is cutting a cake
Tuples	5 a cup sitting on a table	4 an apple and an apple	3 a man cutting a cake
Dep	2 a bird on a table	5 a man holding an apple	2 a person cutting a cake
Const	3 a bird sitting on a table	2 an apple	2 a person cutting a cake
Human	1 a bird is on a plate, with a cup, on a tray	1 a girl posing next to an apple tree	1 a girl cutting into a chocolate cake with a blue knife

Fig. 6. Examples of system output. Numbers denote the relative rank given by AMT participants.

most effective symbol-based models, they yield the highest precision (see Tables 4 and 5). Interestingly, Bag-of-Objects performs better than Tuples on the validation set, but worse than all other models on the test set. Furthermore, in terms of micro-averaged precision, Bag-of-Objects is comparable to Constituency on the validation set. Recall that the test set is less homogeneous than the validation set, it was sampled so as to contain images depicting both common and uncommon object co-occurrences. This suggests that object labels are predictive of verbal predicates when they denote interactions between common objects (e.g., *person* cuts *cake*) or typical actions of objects (e.g., *airplanes fly*), while structural representations based on visual relationships are more effective in encoding interactions between rarer objects. The comparison to the state-of-the-art image representations (VGG)

Table 5. *Macro-averaged precision on 208 COCO test images. Numbers in parentheses are micro-average.*

Model	P@1	P@5	P@10
Bag-of-Objects	8.6 (12.9)	10.5 (12.3)	13.3 (14.3)
Tuples	11.7 (14.3)	13.4 (15.7)	13.6 (15.4)
Constituency	**19.9 (22.9)**	14.2 (15.1)	11.9 (12.9)
Dependency	15.2 (15.7)	15.7 (16.9)	13.9 (15.6)
VGG	19.5 (17.1)	**17.1 (17.4)**	**16.4 (17.1)**

NIC	5 a bed sitting in a bedroom next to a window	6 a man riding a skateboard down a street	5 a piece of cake sitting on top of a white plate
Temp	4 a bed stuffed a teddy bear	4 a person is holding a person	1 a vase is on a dining table
Tuples	3 a bed has a bear	5 a man holding a man	2 a vase sitting on a table
Dep	2 a bear sitting on a bed	3 a man is riding a bicycle	2 a vase sitting on a table
Const	2 a bear sitting on a bed	2 a man is riding a bicycle	2 a vase sitting on a table
Human	1 a teddy bear lying on a bed	1 a group of people in a neighborhood	3 a dinner table is prepared and waiting for the food to arrive

Fig. 7. Examples of system output. Numbers denote the relative rank given by AMT participants.

corroborates this. As we would expect due to the nature of the dataset, VGG outperforms the symbol-based models in all settings except on the test set when considering only the top-ranked image (P@1, Table 5).

6.5 Error Analysis

Image Retrieval We manually inspected the output of various image retrieval systems on the COCO test data. Figure 8 presents examples of typical errors we found. It shows eight query images (first column) and the one-best image retrieved for each system (columns 2 to 5; for Query 6 we show the two-best images).

Our analysis reveals that models which use symbolic representations of images (i.e., the 80 COCO object class names; all models except for VGG) are susceptible to errors made by the object detectors (e.g., Queries 3 and 6 for Tuples, Query 7). Similarly, objects which

are unknown to the detectors result either in detection errors or cannot be captured by the representations at all (e.g., the TREE in Query 1). The effect of object detection is most pronounced in the Bag-of-Objects system which treats all objects equally, lacking any structural information, e.g., based on the spatial relations between objects. For instance, representations for Query 5, which shows two bears walking beside each other, were built on the basis of the objects BEAR, BEAR, CELL_PHONE, and BIRD, which caused Bag-of-Objects to retrieve an image of a bird as its nearest neighbor. Note that the image retrieved by VGG for Query 5 is visually similar, but does not depict bears either. Recall that VGG represents images by their CNN-based feature vectors, and does not directly rely on discrete object detections.

Query 2 illustrates another source of typical errors for the structural approaches: They struggle with images which do not explicitly depict object (inter)actions. VGG is better at retrieving answers which are visually similar to the query, i.e., which depict *visually* similar objects and background scenes. But it has difficulty finding images which capture the same semantic content as object interactions depicted in the query. For example, Query 3 shows a person and a cake, which all models (except Tuples) get right. In VGG's answer image, however, the person is not *cutting* the cake. For Query 5, VGG's answer shows formally dressed people, but neither of them is holding a glass.

We also observe that for the given dataset (a collection $\mathcal{I}$ of 868 images; see Section 5), our relatively few spatial relations may be too fine-grained. For example, Query 1 was represented as a PERSON being `above_veryclose` to a FRISBEE. $\mathcal{I}$, however, only contains an image of a PERSON being `above_close` to a FRISBEE.

Description Generation Figures 6 and 7 provide examples of the systems' output on the description generation task. As observed with the retrieval task, since all systems (except for NIC) use symbolic representations of images, they are affected by errors which are made by the object detectors and propagated to the generated descriptions. Furthermore, since we focus on objects and their interactions and do not recognize attributes or scenes, descriptions produced by our models often contain less detail compared to human authored ones. For example, they do not mention parts of objects or attributes (see Figure 6 first image). Perhaps unsurprisingly, SMT-based models tend to generate more fluent and natural descriptions than Template (see Figure 6 first row, right-most column; Figure 7 first image). Constituency tends to be better than Dependency in handling uncommon object co-occurrences for which it more often produces grammatical descriptions (see first image in Figure 6 left-most column), or verbalizes only parts of the representation.

As far as NIC is concerned, we observed that 33% of the descriptions it generated are duplicates, i.e., they are identical with captions produced for other images. The proportion of duplicates is considerably less for other models, i.e., between 19% (Tuples) and 13% (Templates). An explanation for NIC's high proportion of duplicates may be that it has a generation mechanism similar to retrieval-based captioning systems—it reproduces the captions from the training images which are visually very similar to the target image. We give an example for Query image 4 (which shows a woman holding a wine glass) in Figure 8. Recall that NIC receives the same input as the VGG retrieval model. For Query 4, NIC generated the description *a couple of men standing next to each other*, which turns out to be an accurate description of the image which was retrieved by VGG.

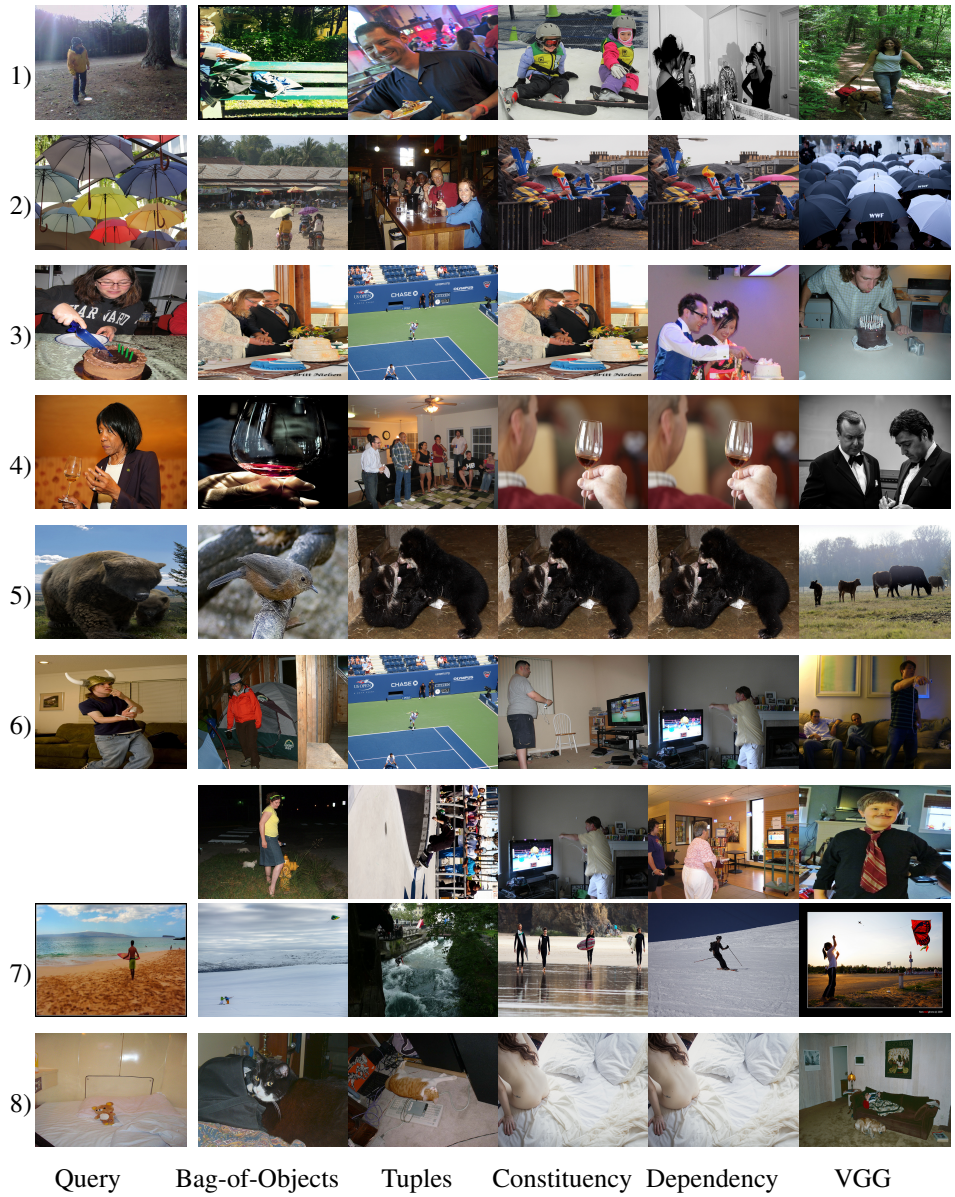


Fig. 8. Query image (first column) and one-best image retrieved from the test data. The second column shows the output of the Bag-of-Objects baseline, the third column shows the output of the Tuples baseline, and so own. For Quey 6, we show the two-best retrieved images.

7 Conclusions

In this paper we focused on the problem of representing the semantics of images. We advocated the use of tree-based representations explicitly encoding the objects detected in images and their spatial relations. We proposed a deterministic procedure for constructing constituent and dependency tree representations and have shown how these can

be used to develop models for image description generation and image-based retrieval. Experimental results demonstrate that tree-based representations are beneficial compared to structure-agnostic, symbol-based models. Constituency- and dependency-based models perform comparably. On balance, the former type of representation might be preferred as it is compositional and able to handle rare objects better. In comparison to models whose immediate inputs are feature representations extracted from a convolutional neural network, our symbol-based models are not empirically stronger but are able to better capture the semantic content of image scenes in terms of participant interactions. An interesting avenue for future work would be to augment our tree-based framework with distributed object representations, or feed NN models with explicit structural information. We would also like to induce structural representations for images while training a model to perform a specific task such as visual question answering (Antol *et al.*, 2015) or image-based story generation (Huang *et al.*, 2016). Finally, we plan to experiment with additional semantic representations such as logical forms (Deng *et al.*, n.d.).

References

[Aditya *et al.*, 2016] Aditya, Somak, Baral, Chitta, Yang, Yezhou, Aloimonos, Yiannis, & Fermuller, Cornelia. 2016. DeepIU: An Architecture for Image Understanding. *In: Proceedings of Advances in Cognitive Systems 2016*.

[Antol *et al.*, 2015] Antol, Stanislaw, Agrawal, Aishwarya, Lu, Jiasen, Mitchell, Margaret, Batra, Dhruv, Lawrence Zitnick, C., & Parikh, Devi. 2015. VQA: Visual Question Answering. *In: The IEEE International Conference on Computer Vision (ICCV)*.

[Chen *et al.*, 2015] Chen, Xinlei, Hao Fang, Tsung-Yi Lin, Vedantam, Ramakrishna, Gupta, Saurabh, Dollr, Piotr, & Zitnick, C. Lawrence. 2015. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv preprint arXiv:1504.00325*.

[Collins & Duffy, 2001] Collins, Michael, & Duffy, Nigel. 2001. Convolution Kernels for Natural Language. *Pages 625–632 of: Advances in Neural Information Processing Systems*.

[Coyne & Sproat, 2001] Coyne, Bob, & Sproat, Richard. 2001. WordsEye: an Automatic Text-to-Scene Conversion System. *Pages 487–496 of: SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*.

[Culotta & Sorensen, 2004] Culotta, Aron, & Sorensen, Jeffrey. 2004. Dependency Tree Kernels for Relation Extraction. *In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*.

[Deng *et al.*, n.d.] Deng, Yuntian, Kanervisto, Anssi, Ling, Jeffrey, & Rush, Alexander M.

[Devlin *et al.*, 2015] Devlin, Jacob, Gupta, Saurabh, Girshick, Ross B., Mitchell, Margaret, & Zitnick, C. Lawrence. 2015. Exploring Nearest Neighbor Approaches for Image Captioning. *CoRR*, **abs/1505.04467**.

[Elliott, 2015] Elliott, Desmond. 2015. *Structured Representation of Images for Language Generation and Image Retrieval*. Ph.D. thesis, The University of Edinburgh.

[Elliott & de Vries, 2015] Elliott, Desmond, & de Vries, Arjen. 2015 (July). Describing Images using Inferred Visual Dependency Representations. *Pages 42–52 of: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

[Elliott & Keller, 2013] Elliott, Desmond, & Keller, Frank. 2013 (October). Image Description using Visual Dependency Representations. *Pages 1292–1302 of: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

[Elliott *et al.*, 2014] Elliott, Desmond, Lavrenko, Victor, & Keller, Frank. 2014 (August). Query-by-Example Image Retrieval using Visual Dependency Representations. *Pages 109–120 of: COLING 2014, 25th International Conference on Computational Linguistics*.

[Girshick, 2015] Girshick, Ross. 2015. Fast R-CNN. *Pages 1440–1448 of: International Conference on Computer Vision (ICCV)*.

[Gupta & Malik, 2015] Gupta, Saurabh, & Malik, Jitendra. 2015. *Visual Semantic Role Labeling*. arXiv preprint arXiv:1505.04474,.

[Heafield *et al.*, 2013] Heafield, Kenneth, Pouzyrevsky, Ivan, Clark, Jonathan H., & Koehn, Philipp. 2013 (August). Scalable Modified Kneser-Ney Language Model Estimation. *Pages 690–696 of: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

[Huang, 2006] Huang, Liang. 2006. Statistical Syntax-directed Translation with Extended Domain of Locality. *Pages 66–73 of: Proceedings of the Association for Machine Translation in the Americas 2006*.

[Huang *et al.*, 2016] Huang, Ting-Hao (Kenneth), Ferraro, Francis, Mostafazadeh, Nasrin, Misra, Ishan, Agrawal, Aishwarya, Devlin, Jacob, Girshick, Ross, He, Xiaodong, Kohli, Pushmeet, Batra, Dhruv, Zitnick, C. Lawrence, Parikh, Devi, Vanderwende, Lucy, Galley, Michel, & Mitchell, Margaret. 2016. Visual Storytelling. *Pages 1233–1239 of: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[Jégou *et al.*, 2008] Jégou, Hervé, Douze, Matthijs, & Schmid, Cordelia. 2008 (Oct.). *Hamming Embedding and Weak Geometry Consistency for Large Scale Image Search - extended version*. Research Report 6709.

[Johnson *et al.*, 2015] Johnson, Justin, Krishna, Ranjay, Stark, Michael, Li, Li-Jia, Shamma, David A, Bernstein, Michael S, & Fei-Fei, Li. 2015. Image Retrieval Using Scene Graphs. *Pages 3668–3678 of: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[Kafle & Kanan, 2016] Kafle, Kushal, & Kanan, Christopher. 2016. Answer-type Prediction for Visual Question Answering. *Pages 4976–4984 of: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Karger *et al.*, 1995] Karger, David R., Klin, Philip N., & Tarjan, Robert E. 1995. A randomized linear-time algorithm to find minimum spanning t rees. *Journal of the ACM*, **42**(2), 321–328.

[Koehn *et al.*, 2007] Koehn, Philipp, Hoang, Hieu, Birch, Alexandra, Callison-Burch, Chris, Federico, Marcello, Bertoldi, Nicola, Cowan, Brooke, Shen, Wade, Moran, Christine, Zens, Richard, Dyer, Chris, Bojar, Ondřej, Constantin, Alexandra, & Herbst, Evan. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *Pages 177–180 of: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*.

[Krishna *et al.*, 2016] Krishna, Ranjay, Zhu, Yuke, Groth, Oliver, Johnson, Justin, Hata, Kenji, Kravitz, Joshua, Chen, Stephanie, Kalantidis, Yannis, Li, Li-Jia, Shamma, David A, Bernstein, Michael, & Fei-Fei, Li. 2016. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations.

[Kruskal, 1956] Kruskal, J. B. 1956. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *In: Proceedings of the American Mathematical Society, 7*.

[Kulkarni *et al.*, 2011] Kulkarni, G., Premraj, V., Dhar, S., Li, Siming, Choi, Yejin, Berg, A. C., & Berg, T. L. 2011. Baby Talk: Understanding and Generating Simple Image Descriptions. *Pages 1601–1608 of: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR '11. Washington, DC, USA: IEEE Computer Society.

[Lan *et al.*, 2012] Lan, Tian, Yang, Weilong, Wang, Yang, & Mori, Greg. 2012. Image Retrieval with Structured Object Queries Using Latent Ranking SVM.

[Li *et al.*, 2011] Li, Siming, Kulkarni, Girish, Berg, Tamara L., Berg, Alexander C., & Choi, Yejin. 2011. Composing Simple Image Descriptions Using Web-scale N-grams. *Pages 220–228 of: Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. CoNLL '11.

[Lin *et al.*, 2015] Lin, Dahua, Fidler, Sanja, Kong, Chen, & Urtasun, Raquel. 2015. Generating Multi-Sentence Lingual Descriptions of Indoor Scenes. *In: Proceedings of the British Machine Vision Conference*.

[Mitchell *et al.*, 2012] Mitchell, Margaret, Han, Xufeng, Dodge, Jesse, Mensch, Alyssa, Goyal, Amit,

Berg, Alex, Yamaguchi, Kota, Berg, Tamara, Stratos, Karl, & Daumé, III, Hal. 2012. Midge: Generating Image Descriptions from Computer Vision Detections. *Pages 747–756 of: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. EACL '12.

[Moschitti, 2006a] Moschitti, Alessandro. 2006a. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. *Pages 318–329 of: Proceedings of the 17th European Conference on Machine Learning*.

[Moschitti, 2006b] Moschitti, Alessandro. 2006b. Making Tree Kernels Practical for Natural Language Learning. *Pages 113–120 of: 11th Conference of the European Chapter of the Association for Computational Linguistics*.

[Och & Ney, 2003] Och, Franz Josef, & Ney, Hermann. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, **29**(1), 19–51.

[Ortiz *et al.*, 2015] Ortiz, Luis Gilberto Mateos, Wolff, Clemens, & Lapata, Mirella. 2015. Learning to Interpret and Describe Abstract Scenes. *Pages 1505–1515 of: Proceedings of the 2015 North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[Palmer *et al.*, 2005] Palmer, Martha, Gildea, Daniel, & Kingsbury, Paul. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, **31**(1), 71–106.

[Papineni *et al.*, 2002] Papineni, Kishore, Roukos, Salim, Ward, Todd, & Zhu, Wei-Jing. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. *Pages 311–318 of: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*.

[Philbin *et al.*, 2007] Philbin, J., Chum, O., Isard, M., Sivic, J., & Zisserman, A. 2007. Object Retrieval with Large Vocabularies and Fast Spatial Matching. *In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Philbin *et al.*, 2008] Philbin, J., Chum, O., Isard, M., Sivic, J., & Zisserman, A. 2008. Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases. *In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Plummer *et al.*, 2015] Plummer, Bryan A., Wang, Liwei, Cervantes, Chris M., Caicedo, Juan C., Hockenmaier, Julia, & Lazebnik, Svetlana. 2015. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. *CoRR*, **abs/1505.04870**.

[Prim, 1957] Prim, R. C. 1957. Shortest connection networks And some generalization. *Bell System Technical Journal*, **36**(6), 1389–1401.

[Roth & Lapata, 2016] Roth, Michael, & Lapata, Mirella. 2016. Neural Semantic Role Labeling with Dependency Path Embeddings. *In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*. To appear.

[Russakovsky *et al.*, 2015] Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, Berg, Alexander C., & Fei-Fei, Li. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, **115**(3), 211–252.

[Schuster *et al.*, 2015] Schuster, Sebastian, Krishna, Ranjay, Chang, Angel, Fei-Fei, Li, & Manning, Christopher D. 2015. Generating Semantically Precise Scene Graphs from Textual Descriptions for Improved Image Retrieval. *Pages 70–80 of: Proceedings of the Fourth Workshop on Vision and Language*.

[Simonyan & Zisserman, 2014] Simonyan, Karen, & Zisserman, Andrew. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, **abs/1409.1556**.

[Uijlings *et al.*, 2013] Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., & Smeulders, A.W.M. 2013. Selective Search for Object Recognition. *International Journal of Computer Vision*.

[Vedantam *et al.*, 2014] Vedantam, Ramakrishna, Zitnick, C. Lawrence, & Parikh, Devi. 2014. CIDEr: Consensus-based Image Description Evaluation. *CoRR*, **abs/1411.5726**.

[Vinyals *et al.*, 2015] Vinyals, Oriol, Toshev, Alexander, Bengio, Samy, & Erhan, Dumitru. 2015. Show and tell: A neural image caption generator. *Pages 3156–3164 of: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Vishwanathan & Smola, 2003] Vishwanathan, S. V. N., & Smola, Alex. 2003. Fast Kernels for String and Tree Matching. *Advances in Neural Information Processing Systems*, **15**.

[Xu *et al.*, 2015] Xu, Kelvin, Ba, Jimmy, Kiros, Ryan, Cho, Kyunghyun, Courville, Aaron, Salakhudinov, Ruslan, Zemel, Rich, & Bengio, Yoshua. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *Pages 2048–2057 of:* Blei, David, & Bach, Francis (eds), *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. JMLR Workshop and Conference Proceedings.

[Yatskar *et al.*, 2016a] Yatskar, Mark, Zettlemoyer, Luke, & Farhadi, Ali. 2016a. Situation Recognition: Visual Semantic Role Labeling for Image Understanding. *In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Yatskar *et al.*, 2016b] Yatskar, Mark, Ordonez, Vicente, & Farhadi, Ali. 2016b. Stating the Obvious: Extracting Visual Common Sense Knowledge. *Pages 193–198 of: Proceedings of the 2016 Conference of the NAACL: Human Language Technologies*.

[Young *et al.*, 2014] Young, Peter, Lai, Alice, Hodosh, Micah, & Hockenmaier, Julia. 2014. From Image Descriptions to Visual Denotations: New Similarity Metrics for Semantic Inference over Event Descriptions. *Transactions of the Association for Computational Linguistics*, **2**.

[Yu *et al.*, 2016] Yu, Licheng, Poirson, Patric, Yang, Shan, Berg, Alexander C., & Berg, Tamara L. 2016. Modeling Context in Referring Expressions. *In: ECCV*.

[Zampogiannis *et al.*, 2015] Zampogiannis, Konstantinos, Yang, Yezhou, Fermller, Cornelia, & Aloimonos, Yiannis. 2015. Learning the Spatial Semantics of Manipulation Actions through Preposition Grounding. *Pages 1389–1396 of: Proceedigs of the IEEE International Conference on Robotics and Automation*.

[Zitnick *et al.*, 2016] Zitnick, C. Lawrence, Vedantam, Ramakrishna, & Parikh, Devi. 2016. Adopting Abstract Images for Semantic Scene Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, **38**(4), 627–638.